

**REMARKS FOR PRE-APPEAL BRIEF REQUEST FOR REVIEW IN U.S. PATENT
APPLICATION NO. 09/882,733 FILED ON 6/15/2001**

Pre-appeal brief review is appropriate in this application, because the rejections in the April 25, 2006 Final Office Action contain clear deficiencies. Morley, Microsoft Computer Dictionary, and Carey, separately, or in combination, fail to disclose data transformation threads or objects within a session thread, as variously recited in independent claims 1, 11, 19, and 29, and thus *prima facie* obviousness required by MPEP §2143.03 has not been established.

Rejection of Claims 1-36 under 35 U.S.C. §103

Claims 1-36 were rejected under 35 USC § 103(a) as allegedly being unpatentable over U.S. Patent Publication No. 2002/10056081A1 (“Morley”) in view of the Microsoft Computer Dictionary 4th Ed (1999), and further in view of U.S. Patent No. 6,078,994 (“Carey”). This rejection is respectfully traversed.

Independent claim 1 recites:

“... *concurrently executing a plurality of data transformation threads within the session thread*, comprising a reader thread that reads data and writes *at least a part of the data* to a first data block buffer; a compressor thread that compresses the part of the data... an encryptor thread and a writer thread.”

Independent claim 11 also recites:

“... *concurrently executing a plurality of data transformation threads within the session thread*, comprising a reader thread... a decryptor thread... and a decompressor thread.”

Independent claim 19 also recites:

“... a reader channel object for reading data and writing *at least a part of the data* to a first data block buffer; a compressor channel object for compressing the part of the data... an encryptor channel object...and a writer channel object... *wherein the*

application executes the reader channel object, the compressor channel object, the encryptor channel object, and the writer channel object concurrently....”

Independent claim 29 also recites:

“... a reader channel object for reading data and writing at least part of the encrypted and compressed data to a first data block buffer; a decryptor channel object... a decompressor channel object... wherein the application executes the reader channel object, the decryptor channel object, and the decompressor channel object concurrently....”

Thus, claims 1, 11, 19, and 29 variously recite execution of multiple data transformation threads, or objects, within one or more session threads. Claims 1, 19, and 29 also variously recite a reader thread, or channel object, that reads (or writes) at least a part of the data to a first buffer.

Accordingly, the transformations (reading/writing, compression, en/decryption, and de/compression)--which conventionally occur in sequence and require the completion of an earlier transformation in the sequence before the commencement of a later process in the sequence--work in parallel as claimed. For example, if a reader thread writes part of the data to a buffer, downstream transformations (e.g., compression by a compression thread) on that part of the data can happen in parallel with the reader thread writing *subsequent* parts of the data to a buffer. Thus, having the threads operate in parallel within a session thread allows the results of one thread to be accessed and used by other threads in the session.

The Examiner admits that the combination of Morley and Microsoft Computer Dictionary does not disclose “concurrently executing a plurality of data transformation threads [or objects] within the session thread.” *See, e.g., pages 5 and 7 of the Final Office Action dated April 25, 2006.* In addition, the combination of Morley and Microsoft Computer Dictionary does

not disclose any of the specifically claimed transformation threads or objects. *See, e.g., pages 4-5 and 6 of the Final Office Action dated April 25, 2006.*

Thus the Examiner relies on Carey to remedy the admitted deficiencies of Morley and Microsoft Computer Dictionary. However, contrary to the Examiner's contentions, Carey also fails to disclose or suggest concurrent execution of a plurality of data transformation threads or objects within one or more session threads. The Examiner uses Carey as evidence that "a multi-threaded model has been well-known to those within the art." *See, e.g., pages 5 and 7 of the Final Office Action dated April 25, 2006.* This statement and the entire Carey reference merely discuss the general use of multi-threading and its advantages over multi-processing. Multi-threading as disclosed by Carey is simply that of different processes running simultaneously within independent threads. But Carey says nothing about multiple operational threads running interdependently within a session thread, such that the results of one thread may be accessed and used by other threads in the session. Carey nowhere discloses or suggests the specifically claimed features of data transformation threads or objects within a session thread, nor the variously claimed transformation threads or objects.

The Examiner asserts that the inventions of claims 1, 11, 19, and 29 are obvious because it would have been obvious to use a multi-thread model for the processes of en/decryption, de/compression, reading and writing. *See, e.g., pages 5 and 7 of the Final Office Action dated April 25, 2006.* However, such reasoning is clearly erroneous. The mere concept of multi-threading is simply too general to disclose or suggest the specifically recited claim limitations, including *execution of multiple transformation threads within a session thread*, and more specifically, concurrent execution of specific transformation threads: *a reader thread, a compressor thread, an encryptor thread, and a writer thread within the session thread*.

Furthermore, because these transformations ordinarily occur in series, merely recognizing the benefits of multithreading in general would not be sufficient to render obvious the claimed limitations. One of skill in the art would have expected these transformations to occur sequentially, and thus would not have expected the operability of a method that executed them concurrently. Further, at best, Carey suggests running multiple threads wherein each thread is an independent and separate transformation, but the results of one thread cannot be accessed or utilized within another thread. By contrast, having the threads operate in parallel within a session thread allows the results of one thread to be accessed and used by other threads in the session.

Even assuming, *arguendo*, that the references could be combined in the way the Examiner describes, the combination would merely suggest a system for distributing compressed and encrypted audio and video data that uses buffers and multiple threads in some capacity. The combination would not disclose executing multiple transformative threads concurrently within a session thread as claimed, and the results of one thread within the session could not be accessed and used by other threads in the session.

To establish *prima facie* obviousness of a claimed invention, all claim limitations must be taught or suggested by the prior art. See MPEP §2143.03. Because Morley, Microsoft Computer Dictionary, and Carey, alone or in combination do not disclose or even suggest data transformation threads or objects within a session thread, nor the specifically claimed transformation threads or objects within a session thread, the deficient disclosures of these references preclude the Examiner from establishing even a *prima facie* basis from which a proper determination of obviousness can be made.

In addition, there is no suggestion anywhere in Morley, Microsoft Computer Dictionary, or Carey as to why one of ordinary skill in the art would be motivated to combine the disclosures of these references. Morley merely discloses encoding, encryption, and compression of audio and video signals, but does not suggest anywhere the desirability of using multiple threads for these transformations. Neither does Carey (or Microsoft Computer Dictionary) mention these transformations as potential uses for multi-threading, nor the benefits of one thread's ability to access the result of other threads in the session. Thus, Morley, Microsoft Computer Dictionary, and Carey fail to suggest the desirability of even the deficient combination described by the Examiner, let alone the more specific limitations of the claimed invention necessary for an obviousness rejection under MPEP §2143.01.

Therefore, it is respectfully requested that the final rejections of independent claims 1, 11, 19, and 29, and their dependent claims 2-10, 12-18, 20-28, and 30-36 be withdrawn.

Respectfully submitted,
Mohan Sankaran et al.

Dated: July 25, 2006

By: /s/ Jennifer R. Bush/

Attorneys for Assignee
Jennifer R. Bush, Reg. No. 50,784
FENWICK & WEST LLP
801 California Street
Mountain View, CA 94041
Phone: (650) 335-7213
Fax: (650) 938-5200